|  |
| --- |
| IADT |
| Developing an OCR System for The Web |
| A literature Review |

|  |
| --- |
| Clemente Gonzales – N00193107  1-7-2023 |

Contents

[1 Introduction to Text Recognition 2](#_Toc123996991)

[2 Optical Character Recognition 3](#_Toc123996992)

[2.1 A Brief Introduction to Optical Character Recognition 3](#_Toc123996993)

[2.2 Overview of The General OCR Algorithm 3](#_Toc123996994)

[2.2.1 Image Acquisition 4](#_Toc123996995)

[2.2.2 Pre-Processing 5](#_Toc123996996)

[2.2.3 Segmentation 5](#_Toc123996997)

[2.2.4 Feature Extraction 6](#_Toc123996998)

[2.2.5 Classification 8](#_Toc123996999)

[2.2.6 Post-Processing 10](#_Toc123997000)

[2.3 A Brief Overview of OCR Libraries 10](#_Toc123997001)

[2.3.1 Tesseract 10](#_Toc123997002)

[2.3.2 OpenCV 11](#_Toc123997003)

[2.3.3 A Comparison of The Libraries 12](#_Toc123997004)

[3 User Experience / Interfaces 12](#_Toc123997005)

[3.1 Introduction to User Experience 12](#_Toc123997006)

[3.2 UX / UI Design Principals 12](#_Toc123997007)

[3.2.1 Aesthetics and Clarity 12](#_Toc123997008)

[3.2.2 Visual Structure / Hierarchy 13](#_Toc123997009)

[3.2.3 Requirements 13](#_Toc123997010)

[3.3 Responsive Design 15](#_Toc123997011)

[3.4 Design Systems 15](#_Toc123997012)

[4 Summary 15](#_Toc123997013)

[References 17](#_Toc123997014)

# Introduction to Text Recognition

A system with the ability to recognize text within natural images have many practical applications that can provide to be useful for society. These systems, for example, can provide to be essential for users who are visually impaired to allow them to go through different environments, such as a small shop, or simple navigation through different roads and streets within the city.

Text recognition can allow for this kind of usage, as text within natural images provide a large amount of information within it, however despite this usage, it is difficult to process text within natural images.

While there are methods to read different characters using machine learning. There are many problems when reading images from text. Font variation, backgrounds, textures, and lighting can provide challenges to the system, and are present within every image (Wang, (2012, November)).

The goal of this research report is to investigate potential methods to help develop a system that can read text from different images, and the methods used to develop them. One of which, is Optical Character Recognition.

# Optical Character Recognition

## A Brief Introduction to Optical Character Recognition

Optical Character Recognition (OCR) is a heavily researched area in the field of machine learning, artificial intelligence, and computer vision (Ranjan, (2021)). The way OCR works is that a piece of software takes an image as an input, and reads the text and sentences within that image, and converts it into digital text, such as ASCII or Unicode (Matei, 2013). This allows the computer to understand the text being presented to it.

There are many applications that can be endearing for the further study of OCR. Companies within a paper-intensive industry can benefit greatly from the use of this kind of software, where a large collection of forms and documents can be found. This can include the legal industry, banking, healthcare, captcha, optical music recognition, and automatic number recognition (Singh, 2012), and can provide great benefits to these industries.

Despite these benefits, there are still limitations. Such examples consist of the speed and accuracy of an OCR program. Many OCR programs will suffer from either being a fast but inaccurate program, or slow but highly accurate one (Matei, 2013). Two examples of this will be looked at later.

This is because OCR software uses Neural Network models to recognize new characters and texts based on previous training. These models are trained from large datasets, consisting of thousands of samples to “learn” how a specific character will look (Ranjan, (2021)).

Due to this, there is a large amount of research in finding ways to improve OCR algorithms to create better accuracy, with lower load times (Matei, 2013).

## Overview of The General OCR Algorithm

Figure 1 displays the general overview of the algorithm. Each of these steps have their own tasks required for the algorithm to predict what the image is representing to text:

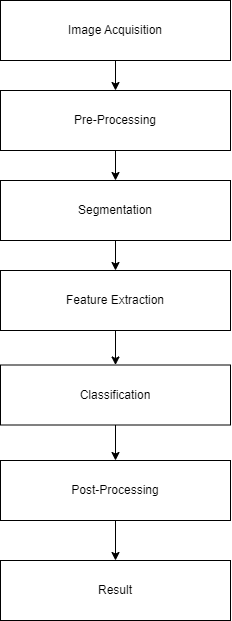


Figure General Overview of the Algorithm

**Image Acquisition:** Is the process of retrieving the image using a hardware system such as a camera from a phone (Ong, 2016). This can consist of a photograph being uploaded to the system and retrieving the image through there, and the image is fed to the system as an input (Ranjan, (2021)). The image should also contain the text that the user wishes to extract, and display on the computer (Matei, 2013).

**Pre-Processing:** Is the steps that are required to enhance the image for segmentation (Ranjan, (2021)). This can consist of the image being changed to a monochrome colour to allow for easier readability (Matei, 2013).

**Segmentation:** This is where the document is segmented into rows, and columns, to extract the words that are within the document (Ong, 2016).

The angles within the segments will be inputted to the neural network, and for the algorithm to confirm it (Matei, 2013).

**Feature Extraction:** Feature Extraction is when the features of different symbols are extracted (Matei, 2013). Symbols are characterized and meaningless features are left out of the algorithm. These extracted features are used to train the system(Mithe, 2013)**.**

**Classification:** The tested image is added to the program for classifying. There are several techniques that can be applied to classification. These methods can include artificial neural networks, support vector machines. During classification, the tested image’s features are compared with the pattern that is in the training dataset (Ong, 2016).

**Post-Processing:** Is the process that helps to improve the accuracy of the recognition. This can consist of the analysis of the syntax and the semantics (Ong, 2016).

Grouping, error detection and correction normally occur within this phase of the process (Eikvil, 1993).

### Image Acquisition

The OCR program requires an input of an image, and within that image containing text, that can be extracted. The image quality and resolution are essential for a clear reading, and an accurate answer (Ranjan, (2021)). Parameters such as colour, fuzziness, lighting, and clarity can affect the reading and answer.

Acquiring the images can come from a variety of sources. These can range from:

1. Photographs: These can come from mobile phones, that contain the required text that the user wishes to extract (Ranjan, (2021)).
2. Scanned Documents: Documents that have already been scanned and stored as an image can be inputted to the system (Ranjan, (2021)).
3. Screenshots and digital images: Mobile phone or computer applications allow users to take screenshots of their screen. This can be used to inserted into the OCR system and be processed the same way as other photographs can be too. Digital images are the same in the sense that they can also be processed through the system to extract text (Ranjan, (2021)).

Through the scanning process, an image of the original document is captured, and optical scanners are used. These optical scanners normally contain a transport mechanism and a sensing device that can allow the system to convert light intensity into grey levels (Eikvil, 1993). This allows the system to read the text in a monochromatic way, ensuring that it remains more accurate than with colour.

It is important to have a clear reading of the image, including the light levels and readability of the text, as the text gets converted into black and white for the program to read. Like less noise, a focus on the text, and lighting should be taken into consideration (Ranjan, (2021)). The result of the reading can be heavily affected if these aren’t taken into consideration.

This process of converting an image to a bilevel image of black and white is often referred to as thresholding. This process is essential as the quality accuracy of the result is dependent on the quality of the bilevel image. The best forms of thresholding are generally ones that are adaptive and can take into consideration the properties of the brightness and the contrast of the image to result in a better and clearer image. These methods however generally are dependent on multilevel scanning and requires much ore computational capacity, resulting in less usage (Eikvil, 1993).

Both papers allude to fact that the image should be clear and readable for the best possible effect, as image processing applications are done using grey scale images, to make the processing efficient (Ranjan, (2021)).

### Pre-Processing

The result of the image that has been scanned through may contain a certain amount of noise, depending on how clear and readable the image is, as the characters may be smudged or broken. Some of these defects can be eliminated using pre-processing, to smooth out these digitized characters (Eikvil, 1993). This process can help optimize the reading and allow for a clearer answer.

During this phase, smoothing and normalization occurs. Smoothing allows rules to be applied to the image, with the help of filling and thinning techniques. Filling removes the small breaks, holes and gaps in the digitized characters, and thinning reduces the width of the line. A technique that is commonly used in smoothing is to move a window across a binary image of that specific character, and while this is there, it applies certain rules to what is inside of the window(Mithe, 2013)**.**

Normalization handles the size, slant, and rotation of the character. For example, if the letter “J” is off centred, and leaning more towards the right, normalization will fix this issue to allow for easier readability and centre it(Mithe, 2013)**.**

The image is also sharpened to enhance the high frequency details. This is to emphasize the edges in the image, to make it easier for the machine to pick out the patterns and letters (Matei, 2013).

The reason for pre-processing is to allow for easier classification for the program to detect the different strokes and classify which belongs to which letter in the alphabet (Ong, 2016).

Another step-in pre-processing is applying adaptive thresholding, which is used to segment the image. This is done by setting the pixels whose values are above a certain threshold to the front value, and all remaining pixels, who do not meet this threshold to the background value (Matei, 2013).

Once pre-processing is complete, it is now ready for the next stage

### Segmentation

Segmentation is the process of locating sections of printed or handwritten text. Segmentation distinguishes text from figures and the image. When segmentation is applied to text, it isolates characters or words that are within that text(Mithe, 2013)**.**Text lines are first segmented, then within these lines of text, the words are then further segmented, and finally from the words, the characters are segmented (Rao, 2016).It is essential to locate the sections of the document where text have been written and identify them from figures and graphics. For example, in the process of automatic mail-sorting, the address must be located and separated from other print on the envelope like the stamps or company logos, before we get to recognition (Eikvil, 1993).

Most OCR algorithms will segment words to isolated characters that gets recognized individually. In general, this form of segmentation is done by isolating each connected component, that is each connected black area. It is easy to implement, however there are some problems that occur if the characters are touching or are broken and consist of multiple parts (Eikvil, 1993). These can lead to the system recognizing a completely different word, or leading to inaccuracies.

According to (Eikvil, 1993), there are a few problems within segmentation, and are as follows:

• Extraction of characters that are touching or fragmented. These distortions can lead to multiple joint together characters being understood as one single character, or that a piece of a character is thought to be an entirely different symbol. These joints can occur if the photograph or document is a dark photocopy or if it is scanned at a low threshold. Joints are also common if the fonts are serifed. If the document has a light photocopy or is scanned at a high threshold, it is also possible for a split to occur due to the circumstances of the scan (Eikvil, 1993).

• Distinguishing noise from text can also be an issue. Dots and accents might be misinterpreted as noise, and vice versa, causing inaccuracies to the finished product (Eikvil, 1993).

• Misinterpreting an image or geometry and seeing them as text. This can lead to non-text being sent to recognition and being unable to read it (Eikvil, 1993).

• Misinterpreting text as an image or geometry. In this scenario, it is possible that the text will not be passed to the recognition stage. This often occurs if characters are attached to graphics.

#### Segmentation methods

Document segmentation is a key pre-processing stage in applying an OCR system. It is the process of classifying a document image into homogeneous zones, i.e., that each zone contains only one kind of information, such as text, a figure, a table, or a halftone image. In many cases, the accuracy rate of systems related to the OCR heavily depends on the accuracy of the page segmentation algorithm used (Rao, 2016).

There are three categories of Algorithms of document segmentation according to (Rao, 2016). These are as follows:

* Top-down methods, which is when a document is segmented from large regions into smaller regions recursively. The segmentation process will stop when it reaches a stage that meets a criterion, i.e., it reaches the final range of segmentation (Rao, 2016).
* Bottom-up methods will search for interest pixels, and groups these interest pixels. They manage the interest pixels into associated elements that represent characters that are combined into words, or lines, or blocks of text afterwards (Rao, 2016).
* Hybrid methods is an integration of both approaches seen from above (Rao, 2016).

### Feature Extraction

Feature extraction is the process of extracting relevant features from objects or alphabets to build a feature vector. The objective of this step is to capture vital traits and symbols (Eikvil, 1993). These vectors are used by classifiers to identify the input unit with the objective output unit. The easier the features are to determine; the more effortless classification becomes (Rao, 2016).

According to (Rao, 2016), there are many methods in which feature extraction can be accomplished. For example, one such method consists of a directional chain code feature, and zoning for handwritten numerical recognition. It consists of a feature vector of length 100 and have a high level of accuracy, however it proves to be time consuming and complex (Rao, 2016).

Another potential method that was seen in (Rao, 2016) is that the end points are the potential features towards recognition. These features use horizontal/vertical strokes and for handwritten numerals obtained a recognition accuracy of 90.50%. Despite this, the method uses the thinning process which can result in a loss of features.

However, according to (Eikvil, 1993), the techniques of feature extraction can be categorized into three different groups, where the features can be extracted. This can consist of:

* The distribution points.
* Transformations and series expansions.
* Structural Analysis

In both (Rao, 2016) and (Eikvil, 1993)**,** we see that noise and distortion can skew the results that are given. (Eikvil, 1993),however, goes into more detail, we can see that noise, distortions, style variation, translation, and rotation can all effect the speed of the recognition, the complexity of implementation and whether the system will need further support to recognize the text. The following table can display an example of feature extraction.

Table

Description automatically generated

Figure Feature and Extraction techniques from (Eikvil, 1993)

#### Distribution Points

As seen in article (Eikvil, 1993)and (Rao, 2016), the distribution points are key features in identifying the essential components of the character. These different techniques can be listed below as:

* Zoning – The rectangle covering the character is divided into different regions and densities of black points (Eikvil, 1993).
* Moments – The moments of black points being chosen as a centre (Eikvil, 1993).
* Crossing - The features are found through the number of times the character shape and the vectors along certain directions are crossed (Eikvil, 1993).
* N-Tuples – The joint occurrence of the foreground and background (black and white) points in a certain order being identified as features (Eikvil, 1993).
* Characteristic Loci – Vertical and Horizontal vectors are generated. The segments of a character that intersects with the vector is used as a feature to describe the character (Eikvil, 1993).
* Transformation and Series expansions - reduce the dimensionality of the feature vector and the extracted features can be made invariant to global deformations (Eikvil, 1993).

#### Structural analysis

Structural analysis allows features that describe the geometric structure of a symbol to be identified and extracted. These can include the physical characteristics of the character, commonly identifying loops, strokes, intersections, lines, bays, and endpoints. Structural analysis is a technique with a large tolerance to style of the character and noise variations but are not very tolerant to rotation and translation (Eikvil, 1993).

### Classification

The classification state is when the features are extracted to identify the text segment that is according to the rules. Classification is generally accomplished by the comparison of feature vectors that correspond to the input characters with the representative of each character class (Lehal, 1999).

Before doing this however, the classifier should possess a range of training patterns (Verma, 2012).There are many classification methods, that have been proposed by different researchers, and some of them will be looked at in a later section. These techniques can consist of statistical methods, template matching, syntactic methods, artificial neural networks, and kernel methods (Verma, 2012).

According to (Lehal, 1999), the nearest neighbour classifier (statistical method as mentioned in (Verma, 2012)) and binary classifier trees have been the two most common classifiers.

#### Statistical methods (K-Nearest Neighbour)

The reason behind using the statistical methods is to determine which category the pattern belongs to. This is done through the observation and measurement process, where it prepares a set of numbers that is used to prepare a measurement vector (Lehal, 1999).

The K-NN rule is a non-parametric recognition technique. The method compares unknown patterns received from the feature extraction stage to a set of patterns that have been labelled with class identities during the training stage. A pattern is then recognized to be of the class of pattern, to which it has the closest distance (Verma, 2012).

This technique is effective for classification problems where the patterns consist of a limited number of variations. For clear and specific machine-printed text, the patterns of each class tend to be clustered tightly towards the patterns that represent that class. The nearest neighbour approach can be effective method of classification; however, it can suffer from memory and size issues if more fonts are added (Lehal, 1999).

Other common statistical methods are Bayesian classification, which assigns a pattern to a class with max posteriori probability, Quadratic Discriminant Function (QDF), Linear Discriminant Function (LDF), Cross Correlation, Regularized Discriminant Analysis (RDA) and Euclidean Distance (Verma, 2012).

#### Artificial neural networks

A neural network is an architecture that contains a massive interconnection of flexible node processors (Rao, 2016). The output from one node reinforces the next one in the network, and a result is garnered from the complex collaboration of all nodes. Feed-forward and feedback neural networks can be considered as a categorization of a neural network architecture (Rao, 2016).

Diagram

Description automatically generated

Figure (Lopez, 2017), Basic ANN Structure

In (Lopez, 2017), we can see the basic structure of an artificial neural network. The network contains of an input, hidden, and an output layer. Training is done using a method known as back-propagation.

In (Eikvil, 1993), we see the use of back propagation, where it mentions that within this kind of network, which consist of many layers of interconnected elements, when a feature vector enters as an input through the input layer, each element of the layer calculates the weighted sum of the input and transforms it into an output done using a non-linear function. While the neural network is training, the weights are adjusted until the required output is achieved. The problem with a neural network in OCR is their limited predictability due to training, however they are very adaptive which provides to be a general advantage. Therefore, libraries such as OpenCV, who uses CNN for OCR is widely used, because of this adaptability.

According to (Verma, 2012),the most used neural networks for OCR and the pattern classification task is the feed-forward network, the Radial-Basis Function (RBF) networks, Convolutional Neural Networks (CNN), Vector Quantization (VQ), Learning Vector Quantization (LVQ), and auto-association networks.

An interesting study that was found in (Matei, 2013), shows that combining the two methods of using K-NN and Neural networks can provide staggering results. Their study uses Artificial Neural Network and K-NN as a confirmation algorithm, combining the two, and bases the vectors on the angles of digits rather than pixels. Some advantages found within their system consist of the ability to work in different light levels and exposure conditions, insensitivity to rotation and being able to deduct and use exploratory character recognition which provided great success with moderate levels of training.

#### Template matching

One of the simpler methods towards pattern recognition, template matching is the approach where a prototype of the pattern that must be recognized is available. The pattern that is to be recognized is compared with patterns that are already stored and ignores the size and style of these patterns (Verma, 2012).

### Post-Processing

Post-Processing is the final stage of the program, where it is used to make corrections towards spelling errors, and the grouping of the words together (Eikvil, 1993).

* **Grouping:** The result of the symbol recognition on a document will result in individual symbols. Grouping will take these symbols and group them together where they each belong with each other, making words, numbers, and sentences. This process is known as grouping, where symbols that are found to be adequately close are grouped together (Eikvil, 1993).
* **Error-detection and Correction:** In advanced OCR problems, a system consisting only of single-character recognition will not be sufficient. Errors will often occur, and a correction must be provided. One of these approaches is done using syntax, where it follows the rules of grammar, for example a capital after a full stop. Another approach is done using dictionaries, where it can provide to be the most efficient method for error detection and correction (Eikvil, 1993).

The next section will now be discussing how the general algorithm is used in OCR libraries.

## A Brief Overview of OCR Libraries

### Tesseract

Tesseract is an open-source OCR engine. Developed in 1984 – 1994 at HP. Tesseract uses the Line Finding algorithm. This algorithm is designed so that a skewed image has the capability to be recognized without having to de-skew the image, which saves the image quality. Blob filtering and line construction is some of the key parts of the process. More details of the process can be found at (Smith, 2007, September), however the process is like what is described in the general overview of the algorithm section above.

In short however, the program first reads the text and converts it into black and white outlines, which are then converted into Blobs. These Blobs are arranged into text lines, where the lines and regions are then analysed for some fixed area or are of equal text size. Text is divided into two words, with these being definite space and fuzzy spaces. Recognition is also started as a two-pass (Patel, 2012).

Diagram

Description automatically generated

Figure Tesseract Architecture (Patel, 2012)

Tesseract can provide better accuracy to grey-scale images rather than coloured ones, seen in the experiment conducted in (Patel, 2012). Tesseract works best when it comes to grey-scale images, and as a result works best when working with these kinds of images, with very high accuracy, however, falls short when it comes to images with colour.

This method can be effective in capturing hand-written documents, where it is a simple monochrome colour, and there is not much on the written paper being presented, using tesseract can provide to be an effective solution.

### OpenCV

Open-Source Computer Vision (OpenCV) is a library with programming functions for real-time computer vision. The library has over 2000 algorithms and has been widely used around the world. Programmers can implement many digital image-processing algorithms for mobile phones (Ma, 2000).

The use of OpenCV can be beneficial in OCR. This is because OpenCV can detect and factor in the different noise levels, and colours that occur when using OCR is regular day-to-day life.

A study in (Goel, 2019), shows a process where it uses Convolutional Neural Networks (CNN) and the OpenCV Library to develop an accurate OCR system. It extracts natural scenes from images, where there can be a wide range of colours, fonts, textures, lighting conditions, etc. and can use the two methods effectively to accurately capture the text from an image (Goel, 2019). From this, it can be safely assumed that it is possible to use OpenCV in open areas, for example a street road, in attempts to examine a sign to see where a user is**,** and is more accurate than using something like Tesseract, despite taking longer to process.

### A Comparison of The Libraries

The reason for discussing the two libraries is to see their different benefits and advantages. From researching the Tesseract, KerasOCR, EasyOCR, and OpenCV, a conclusion can be drawn that there are many benefits for the use of each library and can be highly effective depending on the goal of the software a person is developing needs to accomplish.

Tesseract is highly effective when used in document reading, where the image and font presented is in a monochromatic colour and noticeably clear to read. This can work best with handwritten documents, where the ink and the paper are highly contrasted. Pairing this with the high accuracy, and lower loading times, it can prove to be the most effective method of OCR in a system, where the main goal is to read notes on paper rather than in the outdoors.

OpenCV however, can be used for recognition in the wild instead, with the drawback of slower loading times is much more optimized for OCR in the wild. The capability of this is mentioned above, where it uses CNN to look at many images and interpret the text.

Overall, the two libraries both provide benefits for different goals, and can compensate for each other’s weaknesses in OCR.

# User Experience / Interfaces

## Introduction to User Experience

The concept of User Experience (UX) and User Interface (UI) is a term that is used within many platforms. Specifically in this dissertation, the UX and UI for the web will be reviewed to understand and implement the best user experience possible.

Poorly designed UI can spoil a user’s experience with a website, and to avoid this, research will be done to implement standard design practices and foundations that can be globally applied to many websites.

The goal is to give the user the best potential experience when interacting with the website, as well as be an aesthetically pleasing website to look at, and to retain their attention and consistently keep returning.

The next section will be focused on discussing the UX and UI Design Foundations, to help us understand the reasoning and psychology behind these design principals.

## UX / UI Design Principals

In this section, the essential design principals will be discussed and researched. The steps and goals for each design principal will be looked at here, starting with the look of the website.

### Aesthetics and Clarity

As design and aesthetics and visual composition is attractive to the eye, it is possible to convey the idea of your website quickly and clearly through the design alone. Graphics design principles that include contrast, hierarchy, spacing, alignment, and using colour effectively provides the overall look and feel of an interface (Bhaskar, 2011). These elements are essential when creating a web application, as these concepts can also be applied to web-apps.

Based on previous experience, it is effective to build a design system, where the designer chooses a fitting colour palette that suits the feel of the website, choses the typography which complement the website, and each other, and the style of the icons and buttons to be used to help keep the look of the website consistent.

### Visual Structure / Hierarchy

Visual Structure and Hierarchy is an essential part of designing an interface. A visual hierarchy allows users to organize information into clear categories that can be repeated throughout the website (Fleming, 1998). This allows users to better understand what they are looking at, providing clarity when analysing the body for information. It allows users to separate their relevant goals from irrelevant information that is provided within a body of text (Johnson, 2020).

A screenshot of a computer

Description automatically generated with low confidence

Figure [Ref 25] Visual Hierarchy

Looking at fig. 5, it is easy to better distinguish the relevant information on the text on the right. The use of bold and large font for the title, while using smaller text however still in bold for the subheading shows the main ideas that the article provides.

Relevant size can help communicate the relevance of the information provided, as seen in the previous figure. Large items, position of element, and colour and contrast, can help draw a user’s attention (Fleming, 1998). This can allow for the communication of information to be more easily done and is not simply font size but all the other elements listed above can help draw a user’s attention.

### Requirements

The requirements gather phase is an essential part of the design progress. It finds the needs and the subject matter of the application that is being developed. It is essential to understand what is being developed and the plan the functionalities that the system will have.

There is a plethora of different methods to accomplish the requirements gather phase. Those which can be listed below:

#### User Observations

Observing user behaviour while interacting with the website can provide to be beneficial when planning the design of the application. Through observing their behaviour, it is possible to see what the users do, and what they like and dislike about the application (Stone, 2005). This can allow the developers to discover flaws that is found within the design and allow the developer to make changes where necessary to provide a better end-product overall.

Direct observation is when a developer is watching a user interact with their system and taking notes on any issues that the user is facing. This is normally done on the user’s system (Stone, 2005).It is however easy for a developer to overlook an important issue when doing direct observation, as the process is not recorded, and cannot be studied after it is over. This is where it is possible to instead record a user’s actions for further study.

Indirect observation allows just that, where the user can provide a video recording that allows the developers to watch and analyse the recording to find issues they might deem necessary to fix (Stone, 2005).

Both methods have their own benefits and are excellent ways to find issues within the design to fix for a much better product.

#### Interviews

Interviews can also be conducted when gathering requirements. The interview process is done to find the different features that a user would like to see in an application that the developer is creating. There are generally two types of interviews; a structured interview, where questions are generally pre-written and contains little scope in exploring different topics that arise in conversation, and a flexible interview, where the questions are more broad and can explore and dive deeper into a certain topic within that interview (Stone, 2005). Both provide different benefits. The structured interview can allow developers to gather data on different functionalities on the system of something that has a clearer goal, while the general interview can allow developers to gather more ideas on what kinds of functions a system can have, as well as what the different users would like to see

#### Surveys and Questionnaires

Surveys and questionnaires are similar to interviews, where they can be used to gather information and is flexible and friendly in gathering more precise information (Stone, 2005). The use of surveys can be used to mass gather information as tens to hundreds of people can take part in a short survey. This is good when gathering requirements, as it allows users to show developers what they wish to see within their program.

There are two kinds of questions when creating a questionnaire or a survey. One of which are known as “closed questions”. These consist of closed-ended answers, for example “yes” or “no” (Stone, 2005) and can be used for specific topics that can be answered easily.

The second set of questions are “open questions” where they are open to what the user has to say. These questions can provide rich data from what a user has to say due to the openness of the question (Stone, 2005). For example, this can be used to gather feedback on a certain feature and see how a user would like that feature to be improved upon.

## Responsive Design

As technology continues to evolve, responsive design becomes more and more crucial in the role of web development. Many people will own wide range of screens, where websites might not be optimized for. This problem can be resolved with responsive design.

Responsive web design attempts to mix HTML5 and CSS3 features with new design methodology to website architecture, where it the website will adapt to a browser with varying screen sizes (Gardner, 2011). This allows websites to be flexible and compatible with almost any device and provides to be a great tool for accessibility to a website.

According to (Gardner, 2011), responsive design contains three parts:

* A fluid layout is the use of a flexible grid, that allows a website to scale to the width of a screen / browser. These layouts are responsive and can change to whatever screen it fits. Popular practice can consist of using grid systems which consist of columns, gutters, and rows (Gardner, 2011). A fluid layout can allow for a website to be seen and accessed using many different devices and can potentially lead to more people viewing it
* Images and media being flexible and adapt with the website to the browser that it is being viewed through is also essential. Images should be sized accordingly, depending on the screen size. This can be done using CSS, and it is also possible to keep the resolution by using the “max-width” property in CSS (Gardner, 2011). Doing this however can lead to the website loading slower, as the images are mostly kept in high resolution and can provide potential performance issues.
* Media Queries can be used to address usability issues. A common problem that can occur when attempting to develop for all devices is that components are not as optimized as they could be in the location that they are in. For example, if we take a side nav and translate that to a mobile device from a desktop, the side-nav will shrink in width in the mobile version. To combat this, the use of media queries is available, where it can alter the viewing experience and style depending on the device that it is being viewed upon (Gardner, 2011).

All these concepts regarding design can be streamlined using design systems, which we will look at in the next section.

## Design Systems

A design system is a group of requirements that manages design through the maintenance of consistency and reduction of redundancy (Kumar, 2022). Design systems aim to keep the look of a website to be consistent and follow rules of good practice in graphic design.

These design systems are commonly very consistent in look and can help when developing a web application. They can also prove to be useful, as they are very user friendly and clear to read.

Examples of design systems are Material UI design, Fluent Design Systems, Atlassian, and Polaris. These design systems are well respected and commonly used throughout the web. Material Ui Design for example is used by Google.

# A Comparison of EasyOCR and Tesseract

## Introduction

Before beginning the development of the application to be built, a direct comparison was done to test the accuracy of both EasyOCR and Tesseract. The code for EasyOCR was written in Python, and developed by using Anaconda, and the code for Tesseract was written in JavaScript, more specifically React.js and developed using Visual Studio Code.

This comparison was done to determine the accuracy of both libraries, as both provide great benefits and help to make a decision on which library is more preferable to use. The test will be done with a list of the same images to provide a fair test.

## Comparison of the two

The first image to be examined was found in Google Images, with a quick search of a random image to use of a bank sign. In figure 6, it can be seen that the words detected are accurate, however the spacing is slightly off. Nonetheless EasyOCR provides an accurate reading of the sign shown in the image.

A picture containing text, businesscard

Description automatically generated

Figure 6 Bank of America (EasyOCR Reading)

The same cannot be said however for Tesseract’s reading. As expected, due to the different colour, the Tesseract library struggles with the reading of the letters and displays a very inaccurate result as seen in figure 7.

**Graphical user interface, application

Description automatically generated**

Figure 7 Bank of America (TesseractOCR Reading)

The next image that was then examined was a simple paragraph, again found with a quick Google search. In figure 8, the letters and words detected are highly accurate, as the letters are clear to read and are not distorted in any way. EasyOCR results are seen in figure 8.

Text, letter

Description automatically generated

Figure 8 Paragraph EasyOCR result

The same can also be seen in figure 9, where Tesseract is then used. It is important to note that the use of Tesseract proved to be much faster than using EasyOCR, and provides highly accurate results in this format, as the words are easy to read and very clear to see.

**Text

Description automatically generated**

Figure 9 Paragraph Tesseract Result

The next test was to test the accuracy of the application when reading letters and paragraphs through common household objects, in this case from a swimming pool box. In figure 10, EasyOCR again proves to be highly accurate when reading letters, as it has very little inaccuracies as seen in the figure below.

A picture containing logo

Description automatically generated

Figure 10 EasyOCR Reading of Swimming Pool Box

TesseractOCR also proves a very accurate reading of the paragraph seen in the bottom of the box. Although it cannot read “Tango”, as it is in colour and struggles to read it, the paragraph at the bottom of the pages shows its accuracy as it is a clear, black font and easily seen against a contrasting background.

**Graphical user interface, text, application

Description automatically generated**

Figure 11 TesseractOCR Reading of Swimming Pool Box

Although TesseractOCR struggles with different colours and typography, a test was done with both libraries to see how the two would compare against regular fonts from a novel.

EasyOCR was tested first and can be seen in figure 12. Although it is quite difficult to see, the reading was highly accurate with the use of EasyOCR. This process however took around one to two minutes to load, showing that it takes a very long time to return results.

Text, letter

Description automatically generated

Figure 12 EasyOCR Book Reading

TesseractOCR on the other hand only took around 20 – 30 seconds to read the following paragraph and provided highly accurate results. These results can be seen in figure 13.

**Text

Description automatically generated**

Figure 13 TesseractOCR Book Reading

Both results provided clearer readings when using a black and white background with a clearer to use font. TesseractOCR showed the accuracy it can perform when used in this specific scenario, however from previous examples can struggle when reading things used in the outside world such as signposts, and is where EasyOCR then shines.

A final set of tests were then performed, this time testing different handwritten texts with the use of pens, markers, and pencils.

The first library to be tested was EasyOCR, and it once again proves to be highly accurate in the use of an outside environment, and the results can be seen in figure 14.

Text

Description automatically generated

Figure 14 Handwritten EasyOCR Results

These results are to be expected with EasyOCR as it has shown throughout the testing phase that it is highly accurate, with some mistakes here and there.

TesseractOCR is then used, and again struggles to read some of the results, due to the font being hard to read, as it is handwritten text. Although it can occasionally read some words, it struggles to read handwritten texts and lettering, as it is not incredibly accurate, but can still read some of the presented letters. These results can be seen in figure 15, 16, 17 and 18.

**Graphical user interface, text

Description automatically generated**

Figure 15 TesseractOCR Results for Handwritten Text with a Pencil

**Text

Description automatically generated**

Figure 16 TesseractOCR Results for Handwritten Text with a Pen

**Graphical user interface, text

Description automatically generated**

Figure 17 TesseractOCR Results for Handwritten Text with a Coloured Marker

**Graphical user interface, text

Description automatically generated**

Figure 18 TesseractOCR Results for Handwritten Text with a Black Marker

## Conclusion

From these tests, it’s clear that EasyOCR is the optimal library to use if factoring in accuracy, despite this the benefit of speed provided by TesseractOCR can be show in these tests, as it is far faster than EasyOCR and accurate when presented in the correct circumstances. Tesseract can also be used with React and JavaScript and can provide an easier development process than using EasyOCR.

# Summary

In short, OCR is a character recognition system that can be used for computers to recognize characters from an image. There is a general algorithm that requires to go through the steps of image acquisition, pre-processing, segmentation, feature extraction, classification, and post-processing, where the general algorithm of OCR is discussed.

Different libraries can be used to make this process easier and follows most of the steps with the general algorithms, although would contain some differences. These libraries can work well with certain functionalities, for example Tesseract OCR works well with documents, where the image is generally black and white, while KerasOCR and EasyOCR works well with images with many parameters although slightly slower.

A comparison of Tesseract.js and EasyOCR was conducted to see the benefits of both libraries, and concluded that it would be for the best to use Tesseract.js for the project.

In the UI and UX section of the report, the basic design principals were discussed. These consist of how a developer can design a website and good practices to follow when doing so, including contrast, hierarchy, colour, and structure, following a user-friendly design.

Requirements gathering was also discussed to help developers understand flaws in their design or discover the user’s needs and the different functionalities that the website will contain.

Responsive design was also covered for more compatibility options. Media queries, grid systems and fluid layouts are ways to allow for a website to be more compatible with more than one system, for example accessibility for mobile users.

Finally design systems were briefly covered, with their benefits and examples were given of good design systems.

Overall, the research project allowed for a better understanding and insight in the topic of OCR and discover methods of developing an OCR system for the web, with a brief insight to web-design and compatibility with different devices.

# References

Bhaskar, N. U. (2011). General principles of user interface design and websites. *International Journal of Software Engineering (IJSE),*, 2(3), 45-60.

Eikvil, L. (1993). Optical character recognition. *citeseer. ist.* , psu. edu/142042. html, 26.

Fleming, J. &. (1998). Web navigation: designing the user experience. *Sebastopol, CA: O'reilly.* , p. 166.

Gardner, B. S. (2011). Responsive web design: Enriching the user experience. *Sigma Journal: Inside the Digital Ecosystem*, 11(1), 13-19.

Goel, V. K. (2019). Text extraction from natural scene images using OpenCV and CNN. *Int. J. Inf. Technol. Comput. Sci*, 11(9), 48-54.

Johnson, J. (2020). Designing with the mind in mind: simple guide to understanding user interface design guidelines. *Morgan Kaufmann*.

Kumar, S. (2022, June 29). *10 most popular design systems to learn from in 2022 for UX Designers*. Retrieved from Medium; UX Planet: https://uxplanet.org/10-most-popular-design-systems-to-learn-from-in-2022-for-ux-designers-18a24843a860

Lehal, G. S. (1999). Feature extraction and classification for OCR of Gurmukhi script. *VIVEK-BOMBAY-*, 12(2), 2-12.

Lopez, M. M. (2017). Deep Learning applied to NLP. *arXiv preprint*, arXiv:1703.03091.

Ma, D. L. (2000). Mobile camera based text detection and translation. *Department of Electrical Engg Department Stanford University*.

Matei, O. P. (2013). Optical character recognition in real environments using neural networks and k-nearest neighbor. *Applied intelligence*, 39(4), 739-748.

Mithe, R. I. (2013). Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1), 72-75.

Ong, V. &. (2016). Using k-nearest neighbor in optical character recognition. *ComTech: Computer, Mathematics and Engineering Applications*, 7(1), 53-65.

Patel, C. P. (2012). Optical character recognition by open source OCR tool tesseract: A case study. *International Journal of Computer Applications*, 55(10), 50-56.

Ranjan, A. B. ((2021)). OCR Using Computer Vision and Machine Learning. In Machine Learning Algorithms for Industrial Applications. *Springer, Cham*, (pp. 83-105).

Rao, N. V. (2016). OPTICAL CHARACTER RECOGNITION TECHNIQUE ALGORITHMS. . *Journal of Theoretical & Applied Information Technology*, 83(2).

Singh, A. B. (2012). A survey of OCR applications. *International Journal of Machine Learning and Computing*, 2(3), 314.

Smith, R. (2007, September). An overview of the Tesseract OCR engine. *In Ninth international conference on document analysis and recognition (ICDAR 2007)*, (Vol. 2, pp. 629-633). IEEE.

Stone, D. J. (2005). User interface design and evaluation. *Elsevier*.

Verma, R. &. (2012). A-survey of feature extraction and classification techniques in OCR systems. *International Journal of Computer Applications & Information Technology*, 1(3), 1-3.

Wang, T. W. ((2012, November)). End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st international conference on pattern recognition (ICPR2012). (pp. 3304-3308). IEEE.